1-1-2017

A Graphical Digital Personal Assistant That Grounds and Learns Autonomously

Casey Kennington
Boise State University

Aprajita Shukla
Boise State University

A Graphical Digital Personal Assistant that Grounds and Learns Autonomously

Casey Kennington
Boise State University
1910 University Dr.
caseykennington@boisestate.edu

Aprajita Shukla
Boise State University
1910 University Dr.
aprajitashukla@boisestate.edu

Author Keywords
Grounding; Interactive Dialogue; Personal Assistant

ACM Classification Keywords
H.5.2 User Interfaces: Information Systems

ABSTRACT
We present a speech-driven digital personal assistant that is robust despite little or no training data and autonomously improves as it interacts with users. The system is able to establish and build common ground between itself and users by signaling understanding and by learning a mapping via interaction between the words that users actually speak and the system actions. We evaluated our system with real users and found an overall positive response. We further show through objective measures that autonomous learning improves performance in a simple itinerary filling task.

INTRODUCTION

In spoken interaction, participants signal understanding (e.g., by uttering backchannels) which shapes interaction by allowing conversation participants to know that their interlocutor is understanding what is being said. However, signaling understanding is a challenge for speech-driven agents [6]: some systems display the recognized transcript or utter *okay* after a request has completed, but there is no guarantee that the request was actually understood and could lead to the wrong system action. Moreover, though many systems are based on data-driven robust statistical models, they are generally static in that the models have a predefined ontology and do not continue to improve as they interact with users. Taken together, these system shortcomings are due to a lack of *conversational grounding* which is defined as building mutual understanding between dialogue participants [5]. Our goal is to improve system grounding by signaling backchannels to users in an intuitive way and by autonomously improving the mapping between what users say and system actions.

Our personal assistant (PA) system, which we explain further in Section 3, works incrementally (i.e., it updates its internal state word by word) as it comprehends and gives visual cues of understanding through the GUI, and, crucially, if the system displays incorrect cues, a user can correct the misunderstanding immediately instead of at the end of the request. Because incremental processing lends itself to a system that can signal backchannels, an incremental system can also be more autonomous–requests that have been repaired and confirmed locally can be used as examples on how the system can improve understanding through conversational grounding.

In Section 4 we explain how we evaluate our system with real users under two different settings: a baseline system and a system that learns autonomously. Our user evaluations show that our system is perceived as intuitive and useful, and we show through objective measures that it can autonomously improve through the interactive process. In the following section, we explain how we build off of previous work.

BACKGROUND AND RELATED WORK

Though grounding between systems and users is a challenge [11], we build directly off of recent work that was perceived by users as natural and allowed users to accomplish many tasks in a short amount of time [9] and [13, 4] which addressed misalignments in understanding in a robot-human interaction scenarios. Also directly related is [8] which used a robot that could signal incremental understanding by performing actions (e.g., moving towards a referred object). Backchannels play a role in the grounding process; [14], for example, used prosodic and contextual features in order to produce a backchannel without overlapping with users' speech. We use a GUI to display backchannels (i.e., we need not worry about overlap with user speech).

SYSTEM DESCRIPTION

Incremental Processing

Our system is built as a network of interconnected modules as proposed by the *incremental unit* (IU) framework [15], a theoretical framework for incremental dialogue processing where bits of information are encoded as the payload of IUs; each module performs some kind of operation on those IUs and produces IUs of its own (e.g., a speech recognizer takes audio input and produces transcribed words as incremental units). The IU framework allows us to design and build a personal assistant that can perform actions without delay which is crucial in building systems that can ground with the user by signaling ongoing understanding–an important prerequisite to autonomous learning (explained further below).

It has been shown that human users perceive incremental systems as being more natural than traditional, turn-based systems [1, 17, 16, 3, 9], offer a more human-like

experience [7] and are more satisfying to interact with than non-incremental systems [2]. Moreover, psycholinguistic research has also shown that humans comprehend utterances as they unfold [19, 18].

System Overview

Our system builds directly off of [9], which introduced a system composed of four main components: automatic speech recognition (ASR) which incrementally transcribes spoken utterances, natural language understanding (NLU) explained below, a dialogue manager (DM) using OpenDial [12] which determined when the system should `select` (i.e., fill a slot in a semantic frame), `wait` for more information, `request` confirmation, or `confirm` a confirmation request, and the final component was a GUI, also explained below. Figure 1 conceptually shows these components and how the information (i.e., IUs) flows between them. As our work focuses on improvements made to the NLU and GUI to improve conversational grounding, we explain these two components in greater detail.

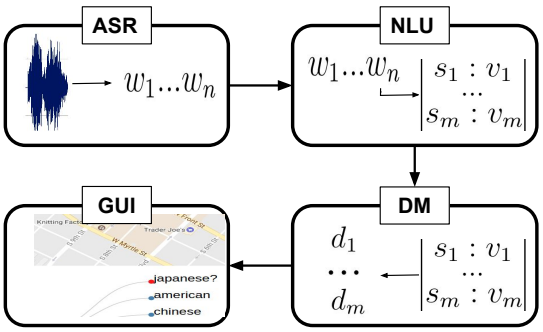

Figure 1. System overview.

Natural Language Understanding

For NLU, we applied the *simple incremental update model* (SIUM) [10] which can produce a distribution over possible slots in a semantic frame. This NLU works incrementally: it updates the distribution over the slot values as it receives words from the ASR. The model is formalized below:

$$P(I|U) = \frac{1}{P(U)} P(I) \sum_{r \in R} P(U|R=r) P(R=r|I) \quad (1)$$

Where $P(I|U)$ is the probability of the intent I (i.e., a semantic frame slot) of the utterance U. R is a mediating variable of *properties*, which maps between aspects of U and I. For example, `italian` is an intent I, which has properties `pasta`, `mediterranean`, `vegetarian`, etc. For training, the system learns a mapping between words in U and the properties in R. For application, for each word in U, a probability distribution is produced over R which is summed over for each I. In our experiments, most properties are pre-defined (which is common), but sometimes properties need to be discovered, e.g., for street names which are unique to an area or city. Our system can discover properties and make use of them without prior training data using a Levenshtein distance calculation between the property and word strings (similar to [9]). As the system interacts with the user, it learns mappings between words and properties autonomously.

Grounded Conversation with an Informative GUI

Our GUI has a map (using the Google Search and Maps APIs), a list of suggestions, and an itinerary created by the user derived from the suggestions. Figure 2 portrays this: the top half of the GUI is a map annotated with the location of suggested items (in this example, restaurants). If a user selects any item in the *Suggestions* list (e.g., by tapping or clicking on it), it is added to the *Itinerary* list for later reference.

Grounding through the GUI: Figure 2 shows the state of the GUI for an example utterance *I'm hungry for some medium-priced Japanese food*. The GUI updated (i.e., by expanding branches showing the options and filling the branch with one of those options, as in `price:medium`), thereby signaling to the user continual understanding (i.e., a backchannel beyond just showing the ASR transcription). Nodes colored in red denote where the user should focus her attention. The system is able to signal a clarification request by displaying ●`japanese?` as a branch of `cuisine`. This informs the user not only that there was misunderstanding, but exactly *what part of the utterance* was misunderstood (in this case, it technically wasn't misunderstanding; rather, the system verified the intent of the user). To continue, a simple *yes* would fill `cuisine` with `japanese` thereby completing the expected information needed for that particular intent type (i.e., restaurant). At that point, the system is as informed as it will be so the user can select from the list of suggestions, ranked by relevance to the request utterance. In the event that a clarification request is answered with *no* (or some other negative response), the question mark goes away and the node is filled again in blue; i.e., ○`japanese`. In addition to the tree, the map incrementally updates as the user's utterance unfolds by displaying a list of suggestions ranked by relevance; the location of those suggested items is further annotated in the map with a relevant icon. As the request unfolds, the number of points of interest shrinks, resulting in a zooming-in effect of the map. Taken together, these visual cues provide several signals of ongoing system understanding to the user.

At any point the user can restart by saying a reserved keyword (e.g., *reset*) and at any point the user can "backtrack" by saying *no* which unfills each slot one by one. For example, in Figure 2, if the user had uttered *I'm hungry for some medium-priced Mexican food* and the system filled `price:medium` and `cuisine:japanese`, the user could say *no* which would result in an expanded `cuisine` slot. This allows users to repair potential misunderstandings locally before the system performs a (potentially wrong) task or produces an incorrect response.

Autonomous Learning: Our system further improves upon previous work by leveraging the GUI to learn as

Figure 2. Our system GUI shows the right-branching tree, a corresponding map, suggestions, and a list of items that the user opted to add to the itinerary.

it interacts. We accomplish this by collecting the words of a completed utterance and corresponding filled slots then informing the NLU that the utterance led to the filled slots–effectively providing an additional positive training example for the NLU. The NLU can then improve its probabilistic mapping between words and slot values; i.e., through updating the sub-model $P(U|R)$ by retraining the classifier with the new information. This is a useful addition because the system designer could not possibly know beforehand all the possible utterances and corresponding intents for all users; this effectively allows the system to begin from scratch with little or no training data. It also allows the system to adapt (i.e., establish common ground) to user preferences as certain words denote certain items (e.g., *noodles* could mean Japanese ramen for one user, or Italian pasta for another). Our system has provisions for providing autonomous learning by updating the NLU using the filled slot values and the utterance when the user selects an item in the *Suggested* list to add it to the *Itinerary*. This allows the system to learn without interrupting the user's productivity with an explicit feedback request.

EXPERIMENT

This section explains a user evaluation performed on our PA. Our PA has provisions for finding information in the following domains: *art galleries*, *bars*, *bus stations*, *museums*, *parking lots*, and *restaurants*. These affordances are clearly visible to the users when they first see the PA GUI. Users interacted with one of two versions of our PA: *baseline* or *autonomous learning*. Both versions were the same in that they discovered possible intents (in this experiment, only `bus station`), applied the same GUI (i.e., the annotated map and right-branching tree) displayed selectable options which are added to an itinerary when selected. The *autonomous learning* version improved as explained above. To allow for greater participation diversity, we made our system available through a web interface and posted the link on various social media sites.

Task & Procedure

Participants were asked to use Chrome Browser on a non-mobile device (e.g., a laptop) with a working microphone. Participants took part in the study completely online by directing their browsers to an informed disclosure about the nature of the study, then instructions were given which are simplified as follows: you have been living in a city for a few months and a (fictitious) friend named Samantha wishes to visit you for a weekend. Use our PA to plan out an itinerary for your friend's visit. We chose Boise, Idaho (U.S.A.) as the location for participants to explore (in future work, we will allow participants to set their language and location).

After reading and agreeing to the instructions, the participants were directed to our PA system with which they interacted in their web browser via speech (we used Google ASR, which works incrementally). At any point, they could add candidate items suggested by our PA into the *Itinerary* list. This constituted phase one. After three minutes, a message popped up, showing their itinerary and a request that they re-create the same itinerary again for another friend. The purpose of this is to see if the system had learned anything about their individual preferences or way of expressing their intent as they recreated their original itinerary.

After they acknowledged the pop-up by clicking *OK*, their itinerary was cleared and they were again able to interact with the system, thereby beginning phase two. They were given another three minutes to complete phase two, for a total of six minutes of interacting with our PA. Afterwards, they were taken to a questionnaire about their experience with our PA, followed by a form for them to enter for a gift card drawing, and finally a debriefing. This task favors a system that can suggest possibilities optimized for breadth; i.e., filling a diverse itinerary. Even though we wish to show how our system can ground autonomously, we opted for this task because it represents a realistic scenario beyond previous work. In total, 15 participants took place in our study and filled out the questionnaire, 8 for the baseline settings and 7 for the autonomous learning setting.

Metrics

We report subjective and objective scores. We report objective measures for the following derived from system logs:

- average length of utterances
- number of items added to the itinerary for the first phase
- fscore between itineraries in the two phases (where the itinerary from phase one is the expected itinerary for phase two)
- number of times the user had to *reset* the GUI
- number of times the user had to backtrack
- number of times the system applied improvements

The subjective scores come from the questionnaires where participants responded using a 5-point Likert scale (the italicized portion is a shortened version of the question that we use in the results table below):

- *like the map* - I liked how the screen showed the map and the assistant at the same time.
- *tree representation* - The assistant "tree" could have been better represented in some other way.
- *worked as expected* - I almost always got the results I wanted.
- *intuitive* - the PA was easy and intuitive to use.
- *noticed it improved* - I had the impression that it was improving.
- *speak/pause* - I didn't know when to speak or pause.
- *system predict better* - It could better predict what I wanted.
- *appeared to understand* - It appeared understand what I said.
- *natural interaction* - I felt that the interaction was more natural than the other personal assistants I have used.
- *fix misunderstandings* - I liked that I could fix misunderstandings quickly.

We hypothesize that the *autonomous* version of the system will result in better results than the baseline system which makes no attempt at learning or improvement. For subjective measures, we hypothesize that the overall experience of both versions will be positive (in fact, as a sanity check, for some questions and measures we expect the scores for both versions to be very similar), but overall the impression that the system improved should be higher for the autonomous learning settings.

Results

Objective

Table 1 shows the objective results as averaged over all. The results show that, in general, the two systems produce similar results, as expected. The important difference is in the fscore, which shows how well the itineraries of the two phases match: the itinerary fscore between the first and second phases for the autonomous system is much higher than it is for the baseline system. We conjecture that this is due to the system learning during the first phase what kinds of items the user added to the itinerary (as illustrated by the average number of improvements done by the autonomous version). During phase two when the users were required to make the same itinerary, the autonomous system had a stronger mapping of utterances and previously selected items, thereby predicting to a small degree what their preferences were (which, as explained above, is a form of grounding).

item	baseline	autonomous
avg utt len*	1.63 (0.69)	2.86 (1.51)
avg # itinerary items	2.5 (3.6)	1.75 (1.78)
avg itinerary fscore	0.04 (0.07)	0.5 (0.5)
avg # reset*	11.6 (26.2)	6.12 (10.2)
avg # no*	2.6 (6.9)	2.18 (3.23)
avg # improvements	0 (0)	6.25 (3.9)

Table 1. Objective results: avg. (std). Asterisks denote items where lower scores are better.

Subjective

Table 2 shows the subjective scores for the questionnaire averaged over all participants with standard deviation in parentheses (questions with an asterisk denote questions where lower scores are better). Overall, the subjective scores do not show a strong preference for either system (a t-test revealed no statistical significance using a Bonrefoni correction of 12); though both systems are rated positively. The users did like that the map directly showed points of interest and they liked the ability to reset at anytime. Though they did not have the impression that the autonomous version was improving while they interacted, they did notice that the autonomous version predicted what they wanted more than the baseline system.

question	baseline	autonomous
I like the map	4.5 (0.7)	4.3 (0.9)
tree representation*	3.1 (1.3)	3.0 (1.6)
worked as expected	3.0 (1.3)	3.0 (1.3)
intuitive	3.1 (0.9)	3.3 (1.2)
I noticed it improved	3.0 (1.0)	2.9 (1.2)
speak/pause*	3.4 (1.4)	3.3 (1.6)
predict better*	3.1 (1.2)	2.5 (1.0)
appeared to understand	3.0 (1.2)	3.3 (1.4)
natural interaction	2.5 (1.2)	2.9 (1.0)
fix misunderstandings	3.4 (1.0)	3.0 (1.4)

Table 2. Subjective results from questionnaires: avg. (std). Asterisks denote questions where lower scores are better.

CONCLUSIONS & FUTURE WORK

The results are positive overall: the system is useful and allows users to fill an itinerary using speech. Users were able to recreate their itineraries with the autonomous system much more accurately than with the baseline system Minimal grounding indeed took place through the GUI by the tree and map, both of which updated incrementally as the users' utterances unfolded, by properties (i.e., ontology) discovery by the system, and by improving the mapping between utterances and properties. For future work, we will leverage our system to autonomously improve the dialogue manager.

REFERENCES

1. Gregory Aist, James Allen, Ellen Campana, Lucian Galescu, Carlos Gallo, Scott Stoness, Mary Swift, and Michael Tanenhaus. Software architectures for incremental understanding of human speech. In *Proceedings of CSLP*, pages 1922—-1925, 2006.
2. Gregory Aist, James Allen, Ellen Campana, Carlos Gomez Gallo, Scott Stoness, and Mary

Swift. Incremental understanding in human-computer dialogue and experimental evidence for advantages over nonincremental methods. In *Pragmatics*, volume 1, pages 149–154, Trento, Italy, 2007.

3. Layla El Asri, Romain Laroche, Olivier Pietquin, and Hatim Khouzaimi. NASTIA: Negotiating Appointment Setting Interface. In *Proceedings of LREC*, pages 266–271, 2014.
4. Joyce Y Chai, Lanbo She, Rui Fang, Spencer Ottarson, Cody Littley, Changsong Liu, and Kenneth Hanson. Collaborative effort towards common ground in situated human-robot dialogue. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, pages 33–40, Bielefeld, Germany, 2014.
5. Herbert H. Clark and Edward F. Schaefer. Contributing to discourse. *Cognitive Science*, 13(2):259–294, 1989.
6. Nina Dethlefs, Helen Hastie, Heriberto Cuayáhuitl, Yanchao Yu, Verena Rieser, and Oliver Lemon. Information density and overlap in spoken dialogue. *Computer Speech and Language*, 37:82–97, 2016.
7. Jens Edlund, Joakim Gustafson, Mattias Heldner, and Anna Hjalmarsson. Towards human-like spoken dialogue systems. *Speech Communication*, 50(8-9):630–645, 2008.
8. Julian Hough and David Schlangen. A Model of Continuous Intention Grounding for HRI. In *Proceedings of The Role of Intentions in Human-Robot Interaction Workshop*, number 1, 2017.
9. Casey Kennington and David Schlangen. Supporting Spoken Assistant Systems with a Graphical User Interface that Signals Incremental Understanding and Prediction State. In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 242–251, Los Angeles, sep 2016. Association for Computational Linguistics.
10. Casey Kennington and David Schlangen. A Simple Generative Model of Incremental Reference Resolution in Situated Dialogue. *Computer Speech & Language*, 2017.
11. Geert-Jan M Kruijff. There is no common ground in human-robot interaction. In *Proceedings of SemDial*, 2012.
12. Pierre Lison. A hybrid approach to dialogue management based on probabilistic rules. *Computer Speech and Language*, 34(1):232–255, 2015.
13. Chansong Lui, Rui Fang, and Joyce Yue Chai. Towards Mediating Shared Perceptual Basis in Situated Dialogue. In *Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, number July, pages 140–149, Seoul, South Korea, jul 2012. Association for Computational Linguistics.
14. Raveesh Meena, Gabriel Skantze, and Joakim Gustafson. Data-driven models for timing feedback responses in a Map Task dialogue system. In *Computer Speech and Language*, volume 28, pages 903–922, Metz, France, aug 2014. Association for Computational Linguistics.
15. David Schlangen and Gabriel Skantze. A General, Abstract Model of Incremental Dialogue Processing. In *Dialogue & Discourse*, volume 2, pages 83–111, 2011.
16. Gabriel Skantze and Anna Hjalmarsson. Towards Incremental Speech Production in Dialogue Systems. In *Word Journal Of The International Linguistic Association*, pages 1–8, Tokyo, Japan, sep 1991.
17. Gabriel Skantze and David Schlangen. Incremental dialogue processing in a micro-domain. *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics on EACL 09*, (April):745–753, 2009.
18. Michael J. Spivey, Michael K. Tanenhaus, Kathleen M. Eberhard, and Julie C. Sedivy. Eye movements and spoken language comprehension: Effects of visual context on syntactic ambiguity resolution. *Cognitive Psychology*, 45(4):447–481, 2002.
19. Michael Tanenhaus, Michael Spivey-Knowlton, Kathleen Eberhard, and Julie Sedivy. Integration of visual and linguistic information in spoken language comprehension. *Science (New York, N.Y.)*, 268(5217):1632–1634, 1995.